Mostrar la salida de ping en tiempo real sobre textarea html5 usando pywebview

```
import webview
import subprocess
import threading
import time

# HTML para el formulario
html = '''
<!DOCTYPE html>
<html>
<head>
    <title>Pinger</title>
    <script type="text/javascript">
        function iniciarPing() {
            var ip = document.getElementById("ip").value;
            document.getElementById("resultado").value = ""; // Limpiar el textarea
            window.pywebview.api.ping(ip);
        }

        function actualizarResultado(linea) {
            var textarea = document.getElementById("resultado");
            textarea.value += linea + "\\n"; // Agregar nueva línea
            textarea.scrollTop = textarea.scrollHeight; // Desplazar hacia abajo
        }
    </script>
</head>
<body>
    <h1>Pinger de IP</h1>
    <input type="text" id="ip" placeholder="Ingrese una dirección IP" required>
    <button onclick="iniciarPing()">Iniciar Ping</button>
    <textarea id="resultado" rows="10" cols="50" readonly></textarea>
</body>
</html>
'''

class Api:
    def ping(self, ip):
        # Ejecutar el comando ping
        process = subprocess.Popen(
            ["ping", ip],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE,
            text=True
        )

        # Leer la salida en tiempo real
        def read_output():
            while True:
                output = process.stdout.readline()
                if output == "" and process.poll() is not None:
```

```
                break
            if output:
                # Enviar la línea de salida a JavaScript
                webview.windows[0].evaluate_js(f'actualizarResultado("{output.strip()}")')
            time.sleep(0.1)

        # Iniciar un hilo para leer la salida
        threading.Thread(target=read_output, daemon=True).start()

if __name__ == '__main__':
    # Crear la ventana web
    webview.create_window('Pinger', html=html, js_api=Api())
    webview.start()
```

Crear un form html y sumar 2 numeros con funcion python

```
import webview

# HTML para el formulario
html = '''
<!DOCTYPE html>
<html>
<head>
    <title>Sumador</title>
    <script type="text/javascript">
        function enviarDatos() {
            var num1 = document.getElementById("num1").value;
            var num2 = document.getElementById("num2").value;
            window.pywebview.api.sumar(num1, num2).then(function(resultado) {
                document.getElementById("resultado").innerText = "Resultado: " + resultado;
            });
        }
    </script>
</head>
<body>
    <h1>Sumador de Números</h1>
    <input type="number" id="num1" placeholder="Número 1" required>
    <input type="number" id="num2" placeholder="Número 2" required>
    <button onclick="enviarDatos()">Sumar</button>
    <h2 id="resultado"></h2>
</body>
</html>
'''

class Api:
    def sumar(self, num1, num2):
        # Convertir los números a enteros y sumarlos
        return int(num1) + int(num2)
```

```
if __name__ == '__main__':
    # Crear la ventana web
    webview.create_window('Sumador', html=html, js_api=Api())
    webview.start()
```